

New insight into 2-community structures in graphs with applications in social networks

Cristina Bazgan^{1,2}, Janka Chlebikova³, Thomas Pontoizeau¹

¹ PSL, Université Paris-Dauphine, LAMSADE UMR CNRS 7243, France

[bazgan](mailto:bazgan@lamsade.dauphine.fr), thomas.pontoizeau@lamsade.dauphine.fr,

² Institut Universitaire de France

³ School of Computing, University of Portsmouth, UK

janka.chlebikova@port.ac.uk

Abstract. We investigate the structural and algorithmic properties of 2-community structure in graphs introduced by Olsen [13]. A 2-community structure is a partition of vertex set into two parts such that for each vertex of the graph number of neighbours in/outside own part is in correlation with sizes of parts. We show that every 3-regular graph has a 2-community structure which can be found in polynomial time, even if the subgraphs induced by each partition must be connected. We introduce a concept of a 2-weak community and prove that it is NP-complete to find a balanced 2-weak community structure in general graphs even with additional request of connectivity for both parts. On the other hand, the problem can be solved in polynomial time in graphs of degree at most 3.

Keywords: graph theory, complexity, graph partitioning, community structure, clustering, social networks

1 Introduction

The problematic around community structures is closely related to the well established research areas of clustering and graph partitioning where similar problems have been studied from different aspects. A good introduction and overview of clustering and partitioning results can be found in [6, 11, 12, 14]. The problems associated with communities are well motivated by current research in social networks such as Facebook, Linkedin, see also Section 2 in [5] for more details about various applications. A standard abstract model for such networks are graphs, in which a community in a graph intuitively corresponds to a dense subgraph. More formally, a community structure is a partition of vertices with some additional constraints such as number of edges between parts or general constraints as connectivity for each part. Therefore the new results for communities may find applications in the areas similar to a graph partition such as parallel-computing, VLSI-circuit design, route planning [8] and divide-and-conquer algorithms [15].

There are several definitions proposed for a community structure in the literature together with some structural and complexity results [3, 4, 9, 13]. The

results in this paper are based on the definition of community structure introduced recently by Olsen [13] which seems to be a natural model for a community in undirected connected graphs. We introduce the concept of a weak community structure in which every member of a community considers itself as a part of the community. We investigate the structural properties of the members of communities for fixed number of two communities in the graphs of maximum degree 3 and present some algorithmic results. The results are further extended for a weak community together with additional constraints such as connectivity or the same size for both parts (balanced partition).

The partition of graphs are intensively studied in the literature with various measures to evaluate their optimality, see for example [1, 6]. In the balanced partition problem, which can be seen as a generalisation of the bisection problem to any given number of parts, the goal is to minimise the number of edges between partitions. It is known that the problem can not be approximated within any finite factor in polynomial time in general graphs and it remains APX-hard even if the maximum degree of the tree is constant [10]. It demonstrates that some graph partitions problems which are related to e.g. balanced communities are hard to solve even for restricted graph classes. This indicates hardness of different problems related to a community structure too, hence any positive results in community structure problems are important to get better understanding of differences between community and partition problems.

The paper is structured as follows. In Section 2 we introduce formally some notation and definitions of problems we study. In Section 3 we show that every 3-regular graph has a 2-community structure which can be found in polynomial time, even if the subgraphs induced by each partition must be connected. In Section 4 we prove that every graph of maximum degree 3 has a balanced weak 2-community structure that can be found in polynomial time, while the problem is NP-complete in general graphs even when both parts should remain connected. Conclusions and open questions are provided in Section 5.

2 Preliminaries

In the paper, all considered graphs are undirected and connected. For any subgraph C of the graph $G = (V, E)$ and a vertex $v \in V$ let $N_C(v)$ (resp. $N_C[v]$) be the set of the neighbours (resp. closed neighbours) of v in C , $d(v)$ be the degree of the vertex v in G . For a partition of V into two parts and $v \in V$ let $d_{in}(v)$ (resp. $d_{out}(v)$) be the number of neighbours in its own part (resp. out of its part). A partition $\{C_1, C_2\}$ of V is connected if the subgraphs induced by C_1 and C_2 are connected. A partition $\{C_1, C_2\}$ of V is balanced if the sizes C_1, C_2 differ by at most 1. A graph is k -regular if every vertex is of degree k , $k \geq 2$. A pendant vertex of G is any vertex of degree 1. A star is a complete bipartite graph $K_{1,\ell}$ for any $\ell \geq 1$.

Now we introduce Olsen's definition of a k -community structure from [13].

Definition 1. A k -community structure for a connected graph $G = (V, E)$ is a partition $\Pi = \{C_1, \dots, C_k\}$ of V , $k \geq 2$, such that $\forall i \in \{1, \dots, k\}, |C_i| \geq 2$, and

$\forall v \in C_i, \forall C_j \in \Pi, j \neq i$, the following holds

$$\frac{|N_{C_i}(v)|}{|C_i| - 1} \geq \frac{|N_{C_j}(v)|}{|C_j|} \quad (1)$$

For a weak k -community structure, (1) is replaced by a “weaker” condition

$$\frac{|N_{C_i}[v]|}{|C_i|} \geq \frac{|N_{C_j}(v)|}{|C_j|}. \quad (2)$$

Notice that a k -community structure is obviously a weak k -community structure since $\frac{|N_{C_i}[v]|}{|C_i|} = \frac{|N_{C_i}(v)|+1}{|C_i|-1+1} \geq \frac{|N_{C_i}(v)|}{|C_i|-1}$, but the opposite is not true (see Fig. 1).

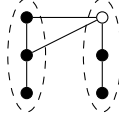


Fig. 1. A 2-weak-community structure of a graph in which the blank vertex does not satisfy condition (1) but satisfies condition (2).

If we remove the restriction k on the number of communities from Definition 1, but (1) is still true for each vertex, we obtain a concept of a *community structure* introduced by Olsen [13]. Olsen proved that a community structure without any restriction on the number of communities can be found in polynomial time for any graphs (with at least 4 vertices) except the star graphs. He also proved that it was NP-complete to find a community structure in a graph in which a given set of vertices is included in a part [13].

In this paper we investigate the problems of a community structure for fixed number of two communities:

2-COMMUNITY

Input: A graph G .

Question: Does G have a 2-community structure?

Obviously, if G has a 2-community structure, it must have at least 4 vertices and not be isomorphic to a star which we assume in the paper.

In the WEAK 2-COMMUNITY problem we are looking for a weak 2-community structure in a graph. Adding the balanced condition to the 2-COMMUNITY problem (for graphs with even number of vertices), we obtain the BALANCED 2-COMMUNITY problem introduced by Estivill-Castro *et al* [9]. Similarly we can define the BALANCED WEAK 2-COMMUNITY problem.

The additional constrain which asks for subgraphs induced by each part of the partition to be connected is a natural constrain useful for the problems related to the connectedness. The CONNECTED 2-COMMUNITY problem is to decide if a graph has a connected 2-community structure, i.e. a 2-community

structure $\{C_1, C_2\}$ such that the subgraphs induced by C_1, C_2 are connected. We can define analogous problems for weak and balanced versions.

The following overview summarises the classes of graphs in which a 2-community structure (hence also weak 2-community) always exists. Depending on the case the results can be extended to connected or balanced communities. All these results are either easy observations or contained as the main results in this paper.

In this way, there always exists a 2-community structure which can be found in polynomial time in (considering graphs with at least 4 vertices):

- 3-regular graphs, even a connected 2-community structure (Section 3).
- graphs of bounded tree-width (except stars), even a balanced 2-community structure. (The results follow from [2], see Section 4 for more details.)
- graphs with minimum degree $\lceil \frac{(c-1) \cdot |V|}{c} \rceil$ where c is the size of an inclusion-wise maximal clique in G . Denote by C such a clique in G and consider the partition $\{C, V \setminus C\}$. Then the condition (1) is satisfied for all vertices in C (the left part of the inequality is 1). The condition (1) is also satisfied for all neighbours $x \in V \setminus C$ of vertices in C since $\frac{d_{in}(x)}{|V|-c-1} \geq \frac{\frac{(c-1) \cdot |V|}{c} - (c-1)}{|V|-c-1} \geq \frac{c-1}{c} \geq \frac{d_{out}(x)}{c}$. Finally, the other vertices in $V \setminus C$ trivially satisfy condition (1) since the right part of the inequality is 0.
- graphs of maximum degree 2 (hence either a cycle or a path). Indeed, any balanced connected partition is a connected 2-community structure (an easy exercise).
- complete graphs, since any partition in which each part has at least 2 vertices is an example of a connected 2-community structure (an easy exercise).

Moreover, there always exists a balanced weak 2-community structure in graphs of maximum degree 3, but this is not true for 2-community structure, see Section 4.

Estivill-Castro *et al* [9] proved that the problem of finding a balanced 2-community structure is NP-complete. In Section 4 we show that the same result also holds for a weak community, even with additional constrain of connectivity for both parts. We also present a shorter proof of the known NP-complete result for a balanced 2-community in general graphs based on an alternative definition of community structure [3], which also implies NP-completeness for a connected balanced 2-community. On the other hand, we prove that every graph of degree at most 3 has a weak balanced 2-community structure which can be found in a polynomial time.

3 2-community structure in 3-regular graphs

In this section we show that any 3-regular graph has a 2-community structure (even connected) computable in polynomial time.

First, the restrictions on the size of partitions are discussed to ensure the vertices fulfil the condition (1) in case of a 2-community structure.

Lemma 1. *Let G be a 3-regular graph of size n . Let $\{C_1, C_2\}$ be a partition of G such that $\lceil \frac{n-1}{3} \rceil \leq |C_1| \leq n - \lceil \frac{n-1}{3} \rceil$. Then each vertex of G which has at most one neighbour out of its own part fulfils the condition (1) of a 2-community structure.*

Proof. Let $\{C_1, C_2\}$ be a fixed partition of G such that $\lceil \frac{n-1}{3} \rceil \leq |C_1| \leq n - \lceil \frac{n-1}{3} \rceil$. It is clear that the condition (1) is true for each vertex which has no neighbour outside its own part. Now let suppose $v_1 \in C_1$ be a vertex which has one neighbour in C_2 and $v_2 \in C_2$ be one which has one neighbour in C_1 . Since $|C_1| \leq n - \lceil \frac{n-1}{3} \rceil$ we have $|C_1| \leq n - \frac{n-1}{3}$ which implies $\frac{2}{|C_1|-1} \geq \frac{1}{n-|C_1|}$. Therefore the condition (1) is fulfilled for v_1 . Similarly since $\lceil \frac{n-1}{3} \rceil \leq |C_1|$ we have $\frac{2}{n-|C_1|-1} \geq \frac{1}{|C_1|}$, hence the condition (1) is fulfilled for v_2 . \square

Lemma 2. *Let G be a 3-regular graph of size n . Let $\{C_1, C_2\}$ be a 2-partition of G such that $|C_1| = \lceil \frac{n}{3} \rceil$ or $|C_1| = \lfloor \frac{n}{3} \rfloor$. Then each vertex of degree 3 from C_1 which has two neighbours in C_2 fulfils the condition (1).*

Theorem 1. *Every 3-regular graph has a 2-community structure. Moreover it can be found in polynomial time.*

Proof. Let $G = (V, E)$ be a 3-regular graph of size n . The algorithm runs in two stages.

Stage 1: The algorithm finds a partition $\{C_1, C_2\}$ of V such that $|C_1| = \lceil \frac{n-1}{3} \rceil$ and at most two vertices from C_1 have more than one neighbour in C_2 .

Stage 2: The algorithm moves some vertices between C_1 and C_2 until $\lceil \frac{n-1}{3} \rceil \leq |C_1| \leq n - \lceil \frac{n-1}{3} \rceil$ and each vertex of G has a restricted number of neighbours out of its own part in a such way that Lemma 1 or 2 can be applied.

Stage 1:

Let $u, v \in V$ be such that $(u, v) \in E$ and put $C_1 = \{u, v\}$. Now repeat the following steps (S1) and (S2) until $|C_1| = \lceil \frac{n-1}{3} \rceil$:

- (S1) Let w be a neighbour of u (or v) which is not in C_1 , put $C_1 := C_1 \cup \{w\}$, $u := w$ (or $v := w$).
- (S2) If there is no such vertex w , the degree of each vertex in the subgraph induced by C_1 is 2 or 3. In such a case let u be any vertex of degree 2 in the subgraph induced by C_1 .

It is clear that at the end of the first stage the algorithm finishes with a set C_1 such that $|C_1| = \lceil \frac{n-1}{3} \rceil$. If it is not possible to apply (S1) and (S2) and $|C_1| < \lceil \frac{n-1}{3} \rceil$ then all vertices in C_1 must have all neighbours in C_1 which means that G is not connected.

Furthermore at most two vertices (u and v) from C_1 may have more than one neighbour outside C_1 and the subgraph induced by C_1 is connected. Define $C_2 = V \setminus C_1$.

Stage 2: We distinguish two major cases:

Case 1: If $\forall w \in C_2, d_{out}(w) \leq 1$ then all vertices in G except u and v have at most one neighbour out of its part. Using Lemma 1, these vertices fulfil the

condition (1). Moreover, $\lceil \frac{n-1}{3} \rceil$ equals $\lfloor \frac{n}{3} \rfloor$ or $\lceil \frac{n}{3} \rceil$ and according to Lemma 2, (1) is also true if u, v have two vertices out of C_1 . Hence, $\{C_1, C_2\}$ is a 2-community structure.

Case 2: There exists a vertex $w \in C_2$, such that $d_{out}(w) \geq 2$.

Now we distinguish several subcases:

- (A) $\forall x \in C_1, d_{out}(x) \leq 1$
- (B) All vertices from C_1 which have more than one neighbor outside C_1 are adjacent to w (only $u, v \in C_1$ are possible candidates).
- (C) No vertex from C_1 which has more than one neighbor outside C_1 is adjacent to w (only $u, v \in C_1$ are possible candidates).
- (D) Both vertices $u, v \in C_1$ have more than one neighbor outside C_1 , but only one of them is adjacent to w .

Case 2(A): Repeat the update step until it is possible:

if $\exists z \in C_2, d_{out}(z) \geq 2$, update C_1 and C_2 as follows: $C_1 := C_1 \cup \{z\}, C_2 := C_2 \setminus \{z\}$.

If no update step is possible, then return $\{C_1, C_2\}$.

After each update step, two neighbours of z in C_1 have degree three. After repeating the update step k times, C_1 has $\lceil \frac{n-1}{3} \rceil + k$ vertices and at least $2k$ vertices in C_1 have degree three in the subgraph induced by C_1 . Hence, we can repeat the update step at most $\lceil \frac{n-1}{3} \rceil - 1$ times. Otherwise the degree of each vertex in C_1 is three which implies G is not connected. Thus $|C_1| \leq 2\lceil \frac{n-1}{3} \rceil - 1 \leq n - \lceil \frac{n-1}{3} \rceil$. Note that now every vertex in G has at most one neighbour out of its part and thus applying Lemma 1, $\{C_1, C_2\}$ is a 2-community structure.

Case 2(B): Wlog we suppose that $d_{out}(u) = 2$ and u is adjacent to w . Furthermore, if $d_{out}(v) = 2$ then also v is adjacent to w .

Now using C_1, C_2 we define a 2-community partition. After the initial update: $C_1 := C_1 \cup \{w\}, C_2 := C_2 \setminus \{w\}$, all vertices in C_1 have at most one neighbor in C_2 and $|C_1| = \lceil \frac{n-1}{3} \rceil + 1$. Then repeat the update step:

- if $\exists z \in C_2, d_{out}(z) \geq 2$, define $C_1 := C_1 \cup \{z\}, C_2 := C_2 \setminus \{z\}$,
- until $|C_1| = 2\lceil \frac{n-1}{3} \rceil - 1$ or if there is no such a vertex z to update C_1 .

There are two possible scenarios:

(i) if $|C_1| \leq 2\lceil \frac{n-1}{3} \rceil - 1$ and there is no such vertex $z \in C_2$ such that $d_{out}(z) \geq 2$, then each vertex in G has at most one neighbour out of its own part. Obviously, $|C_1| \leq n - \lceil \frac{n-1}{3} \rceil$ and due to Lemma 1, $\{C_1, C_2\}$ is a 2-community structure.

(ii) $|C_1| = 2\lceil \frac{n-1}{3} \rceil - 1$ and $\exists z \in C_2$ such that $d_{out}(z) \geq 2$: the update step has been repeated $\lceil \frac{n-1}{3} \rceil - 2$ times and in each step the number of vertices $x \in C_1$ with $d_{out}(x) = 1$ is decreased by at least 2 (the neighbours of z in C_1). It means every vertices in V has all neighbours in its own part, except at most three vertices, each having one neighbour out of its part.

- If $n \equiv 2 \pmod 3$, then the size of $|C_2| = n - (2\lceil \frac{n-1}{3} \rceil - 1) = \lceil \frac{n}{3} \rceil$. Because $|C_1| \leq n - \lceil \frac{n-1}{3} \rceil$, due to Lemma 1, all vertices with at most one neighbor out

of its own part fulfil the condition (1). If a vertex of C_2 is adjacent to exactly two vertices from C_1 then the condition (1) is true according to Lemma 2. A vertex of C_2 cannot be adjacent to all three vertices in C_1 , otherwise $C_1 \cup \{w\}$ is a disconnected part of G . Hence, $\{C_1, C_2\}$ is a 2-community partition.

- If $n \equiv 0 \pmod 3$ or $n \equiv 1 \pmod 3$, then define the last update: $C_1 := C_1 \cup \{z\}$, $C_2 := C_2 \setminus \{z\}$. Now only one vertex of C_1 has one neighbour in C_2 . Because $|C_1| = 2\lceil \frac{n-1}{3} \rceil \leq n - \lceil \frac{n-1}{3} \rceil$, the condition (1) is true for all vertices of G because of Lemma 1. Hence the updated partition $\{C_1, C_2\}$ is a 2-community structure.

Case 2(C): Wlog we suppose that $d_{out}(u) = 2$. Update $C_1 := C_1 \cup \{w\} \setminus \{u\}$, $C_2 := V \setminus C_1$.

Notice that after the update, $|C_1| = \lceil \frac{n-1}{3} \rceil$ and there may be at most two vertices in C_1 which have two neighbours in C_2 . Hence we are again in one of the cases (A)-(D) of second stage, but each time we apply this update the size of the cut between C_1 and C_2 is decreases by two. Therefore the process is finite.

Case 2(D): Wlog suppose that u is adjacent to w and $d_{out}(u) = 2$. Update $C_1 := C_1 \cup \{w\} \setminus \{v\}$, $C_2 := V \setminus C_1$.

Notice that after the update, $|C_1| = \lceil \frac{n-1}{3} \rceil$. Moreover, u has two neighbours inside C_1 since u is obviously not adjacent to v . Hence we are in one of the previous cases of stage 2, since there is at most one vertex in C_1 (the neighbour of v) which could have two neighbours in C_2 . \square

Now we investigate the problem of finding a 2-community structure with additional condition of connectivity for each part. Using the algorithm from Theorem 1 as a tool we extend the result for a connected 2-community structure in 3-regular graphs, but with many fine details in the proof.

Lemma 3. *Let G be a 3-regular graph and $\{C_1, C_2\}$ a connected 2-partition of G with $\lceil \frac{n-1}{3} \rceil \leq |C_1| \leq n - \lceil \frac{n-1}{3} \rceil$ such that each part has at most one vertex with two neighbours out of its own part. Then G has a connected 2-community structure which can be found in polynomial time.*

Proof. The main idea is to move selected vertices between two parts in such a way that it preserves connectivity and offers the options to use Lemmas 1 or 2.

We discuss four cases depending on which vertices have two neighbours out of its own part. Notice that transferring a vertex which has two neighbours out of its part does not compromise the connectivity of the partition.

- (a) If there is no vertex in C_1 and C_2 with two neighbours out of its own part, then using Lemma 1 the partition $\{C_1, C_2\}$ is already a connected 2-community structure.
- (b) If the only vertex with two neighbours out of its own part is in C_2 , then update C_1, C_2 using the following loop:

while $|C_1| < n - \lceil \frac{n-1}{3} \rceil$ and there exists a vertex z in C_2 which has two neighbours in C_1 , update $C_2 := C_2 \setminus \{z\}$, $C_1 := C_1 \cup \{z\}$.

Obviously after each run of the while loop both parts of the partition remains connected. At the end of the while loop

- if $|C_1| < n - \lceil \frac{n-1}{3} \rceil$, then all vertices in G have at most one neighbour out of their parts and satisfy the properties of 2-community structure due to Lemma 1,
 - if $|C_1| = n - \lceil \frac{n-1}{3} \rceil$ then $|C_2| = \lceil \frac{n-1}{3} \rceil$ and hence all vertices in C_2 with two neighbours out of the own part satisfy the properties of 2-community structure due to Lemma 2, the rest of vertices satisfy Lemma 1.
- (c) The only vertex with two neighbours out of its own part is in C_1 . Then the case is similar to (b) by symmetry swapping the roles between C_1 and C_2 .
- (d) There are two vertices:
- $v_1 \in C_1$ with two neighbours in C_2 and let v_1^0 be the neighbour of $v_1 \in C_1$;
 - $v_2 \in C_2$ with two neighbours in C_1 and let v_2^0 be the neighbour of $v_2 \in C_2$.

Now we need to distinguish two cases:

- (i) If $(v_1, v_2) \in E$, then we update the partition as follows. If $|C_1| < n - \lceil \frac{n-1}{3} \rceil$ then define a new partition $C_1 := C_1 \cup \{v_2\}$; $C_2 := C_2 \setminus \{v_2\}$, otherwise $C_1 := C_1 \setminus \{v_1\}$; $C_2 := C_2 \cup \{v_1\}$. Obviously, $\{C_1, C_2\}$ is a connected partition which fulfil initial conditions of lemma, so we can apply case (a), (b), (c) or (d) again. Notice that the case (d) can be repeated only finite number of times since the cut size between C_1 and C_2 decreases each time the case is applied.
- (ii) If $(v_1, v_2) \notin E$ we define the following update $C_1 := C_1 \cup \{v_2\} \setminus \{v_1\}$, $C_2 := C_2 \cup \{v_1\} \setminus \{v_2\}$. The new partition is a connected partition with no change in sizes and the following options are possible:
- If $d_{out}(v_1^0) = 0$ or $d_{out}(v_2^0) = 0$ before the update, then update removes at least one of the vertices with two outgoing edges. Now we can again apply one of cases (a), (b) or (c) which leads to a connected 2-community structure.
 - If $d_{out}(v_1^0) > 0$ and $d_{out}(v_2^0) > 0$ then we can apply case (d) again. This process is finite because each time the size of the cut size between C_1 and C_2 is decreased by 2.

Obviously, the whole procedure can be run in polynomial time. \square

Theorem 2. *Every 3-regular graph has a connected 2-community structure. Moreover it can be found in polynomial time.*

Proof. The algorithm runs in two stages similarly to the algorithm in Theorem 1.

Stage 1: The algorithm finds either a connected partition $\{C_1, C_2\}$ such that $|C_1| = \lceil \frac{n-1}{3} \rceil$ and at most two vertices from C_1 have two neighbours in C_2 or ends up with a connected 2-community structure.

Stage 2: Apply directly Stage 2 from Theorem 1.

The difference to the approach from Theorem 1 is that C_2 remains connected until the end of the first stage, where C_1 is connected in both approaches.

Then we apply the second stage of the algorithm from Theorem 1. Since moving a vertex which has 2 neighbours in the other part never disconnect

any part and all transfers only affect such vertices, the final partition $\{C_1, C_2\}$ remains connected at the end of the second stage.

Stage 1: (for a connected partition)

Choose any vertices $u, v \in V$ such that $\{u, v\} \in E$ and the subgraph induced by $V \setminus \{u, v\}$ is connected. Label the vertices u, v and define $C_1 := \{u, v\}$, $C_2 := V \setminus \{u, v\}$.

The initial construction:

While $|C_1| < \lceil \frac{n-1}{3} \rceil$ and one of the updates (S1), (S2) (in this order) can be applied do:

- (S1) If there exists a vertex $x \in C_2$ such that $d_{out}(x) = 2$, then update $C_1 := C_1 \cup \{x\}$, $C_2 := C_2 \setminus \{x\}$. If all labelled vertices have three neighbours in C_1 , then removes all labels and label one vertex in C_1 which has one neighbour in C_2 .
- (S2) If there exists a vertex $x \in C_2$ such that x is a neighbour of a labelled vertex w in C_1 and $C_2 \setminus \{x\}$ remains connected then update $C_1 := C_1 \cup \{x\}$, $C_2 := C_2 \setminus \{x\}$, label the vertex x and remove label from w . If all labelled vertices have three neighbours in C_1 , then removes all labels and label one vertex in C_1 which has one neighbour in C_2 .

Obviously after each update we can have at most two labelled vertices in C_1 .

Now there are two possibilities how the initial construction can finish:

- (1) The algorithm finishes with $|C_1| = \lceil \frac{n-1}{3} \rceil$. Due to the properties of the construction, the partition $\{C_1, C_2\}$ is connected and at most two vertices from C_1 may have two neighbours in C_2 . In such a case we can move directly to Stage 2.
- (2) If none of the updates (S1), (S2) can be applied and $|C_1| < \lceil \frac{n-1}{3} \rceil$ then we redefine the partition $\{C_1, C_2\}$ using the major update construction to obtain a new partition which leads to a connected 2-community structure.

The major update construction:

Step A: Let $N \subseteq C_2$ be the set of neighbours of all labelled vertices from C_1 , hence $1 \leq |N| \leq 4$. In the first part we define the subsets Q, Q', Z of C_2 and the vertices $q, q' \in N$ (q, q' are not necessarily distinct) such that $\{Q, Q', Z, \{q\}, \{q'\}\}$ is a connected partition of the graph induced by C_2 and each vertex from $Q \cup Q'$ has at most one neighbour in C_1 . Furthermore, the entire set Q (resp. Q') has exactly one neighbour outside Q (resp. Q') in C_2 and this is the vertex $q \in N$ (resp. $q' \in N$).

We show that such Q, Q', Z exist and are always connected. Consider a vertex $v_1 \in N$. The vertex v_1 has necessarily two neighbours in C_2 and the subgraph induced by $C_2 \setminus \{v_1\}$ is disconnected (otherwise (S1) or (S2) from Stage 1 could be applied). Let C_2^1 and C_2^2 be the two connected components of $C_2 \setminus \{v_1\}$.

Define $Q := C_2^1$, $Q' := C_2^2$ and $q = q' = v_1$. Moreover, $Z = \emptyset$ is trivially connected. Hence in case $|N| = 1$ we can now move directly to Step B.

If $|N| > 1$, select another vertex $v_2 \in N$. Such vertex must be in Q or Q' defined above. Consider wlog that $v_2 \in Q$. If $\{v_1, v_2\} \in E$, then update

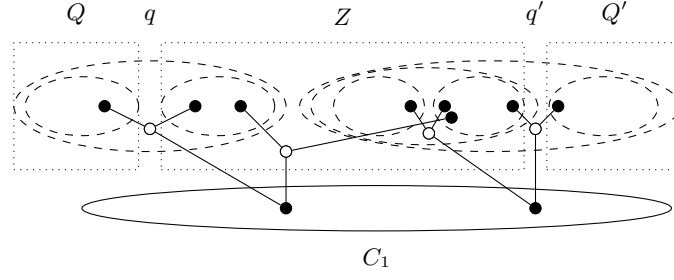


Fig. 2. Splitting C_2 when $|N| = 4$ (vertices in N are blank)

$Q := Q \setminus \{v_2\}$, $q := v_2$, $q' := v_1$ and $Z := \emptyset$. Notice that Q is still connected since v_2 has only one neighbour in Q . If $\{v_1, v_2\} \notin E$, $Q \setminus \{v_2\}$ must be disconnected into two parts Q^1 and Q^2 . Name Q^2 the set which contains a neighbour of v_1 (Q^1 obviously cannot have a neighbour of v_1 , since the other two neighbours are in Q' and C_1 , respectively). Update $Q := Q^1$, $q := v_2$ and $Z := Q^2$. Hence in case $|N| = 2$, the construction in Step A is over and we can continue directly with Step B. Indeed, Q has only $v_2 \in N$ as a neighbour in $C_2 \setminus Q$ and similarly for Q' and $v_1 \in N$.

Suppose now that $|N| \geq 3$. Then select another vertex $v_3 \in N$, the vertex v_3 must be in Q , Q' or Z defined above. If $v_3 \in Z$, then Q , Q' , Z , q and q' already satisfy the properties. Otherwise, $v_3 \in Q \cup Q'$ and wlog we suppose $v_3 \in Q$. If $\{v_3, v_2\} \in E$, then update $Q := Q \setminus \{v_3\}$, $q := v_3$ and $Z := Z \cup \{v_2\}$ which is trivially connected. Otherwise, $Q \setminus \{v_3\}$ must be disconnected, let Q^1 and Q^2 be its connected parts. Denote Q^2 the set which has a neighbour of v_2 (then Q^1 cannot have a neighbour of v_2). Update $Q := Q^1$, $q := v_3$, $Z := Z \cup Q^2 \cup \{v_2\}$. Again, if $|N| = 3$, the construction is over. Indeed, Q has only $v_3 \in N$ as a neighbour in $C_2 \setminus Q$ and there are no changes for Q' . Moreover, Z is connected. Hence in case $|N| = 3$ we can move to Step B. If $|N| = 4$, the construction is similar to the discussion for $|N| = 3$ (see Fig. 2 as an example for $|N| = 4$).

It is also important to notice that if $|N| > 2$ then following the construction $N \setminus \{q, q'\} \subseteq Z$.

Step B: Using the sets Q , Q' , Z and vertices q and q' we define a new partition of V . Then the step B only consists in looking at the size of the sets $Z, Q, Q', Z \cup Q \cup \{q\}, Z \cup Q' \cup \{q'\}$ and update $\{C_1, C_2\}$ depending on the size of those sets which are known to be connected.

The detail of Step B is based on awkward discussion of several cases. \square

4 Balanced 2-community structure

In this section we first prove that every graph of maximum degree 3 has a balanced weak 2-community structure that can be found in polynomial time. On the other hand, we show that the BALANCED WEAK 2-COMMUNITY problem is NP-complete in general graphs similarly to the BALANCED 2-COMMUNITY problem.

The latter result was presented as the main result in [9] and here an alternative shorter proof is given. Both NP-complete results are valid if connectivity of both parts is required.

Remark. As follows from Section 3, every 3-regular graph has a 2-community structure. But if we look for a balanced partition there exists a 3-regular graph which doesn't have a balanced 2-community structure, see Fig. 3. The graph is obtained by linking three “cross gadgets”. First notice that if a 2-community exists for the graph, then all vertices of each cross gadget must be in the same part. Indeed, each vertex of such community structure must have two neighbours in its own part since we search a 2-community structure. But on the other hand, it is impossible to divide the graph into two balanced parts without splitting a cross gadget.

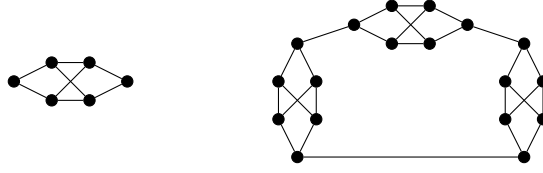


Fig. 3. A cross gadget and a 3-regular graph with no balanced 2-community structure.

Nevertheless, if we focus on a weak community, we can prove that BALANCED WEAK 2-COMMUNITY has always a solution in graphs of maximum degree 3. Moreover, a balanced weak 2-community can be found in polynomial time in such graphs.

Theorem 3. *Any graph of maximum degree 3 with at least 4 vertices has a balanced weak 2-community structure. Moreover, such a community structure can be found in polynomial time.*

Proof. Let $G = (V, E)$ be a connected graph of maximum degree 3. First notice that in any balanced partition, any vertex of degree 1 fulfils condition (2), even if its neighbour is not in its own part. Then, the only vertices which may not satisfy this condition are vertices of degree 2 or 3 which have no neighbour in their own part.

Moreover, for any balanced partition of G , every vertex of degree 2 or 3, which has at least one neighbour in its own part, satisfies condition (2).

Choose any balanced partition $\{C_1, C_2\}$ of G . Then repeat the following steps until the partition is a weak 2-community structure:

1. If every vertex of degree 2 or 3 has at least one neighbour in its own part, then return $\{C_1, C_2\}$ as a weak 2-community structure.
2. If there exists one vertex of degree 2 or 3 in both parts which has no neighbour in its own part, then swap these two vertices.

3. If there is only one partition that contains a vertex v of degree 2 or 3 which has no neighbour in its own part (wlog suppose $v \in C_1$), then choose a vertex $w \in C_2$ such that w has at least one neighbor in C_1 and update: $C_1 := C_1 \cup \{w\} \setminus \{v\}$, $C_2 := C_2 \cup \{v\} \setminus \{w\}$.

First notice that if case 3 occurs, such vertex w always exists since the graph is connected. Moreover, in cases 2 and 3, notice that the partition is still balanced. Besides, the size of the cut between partitions C_1, C_2 always decreases (by at least 4 in case 2, by at least 2 in case 3) so after finite numbers of iteration (bounded trivially by $O(|V|^2)$), only case 1 remains. Hence at the end of the loop, the algorithm returns a balanced weak 2-community structure. \square

It can be observed that the BALANCED 2-COMMUNITY problem (hence also BALANCED WEAK 2-COMMUNITY) is polynomially solvable for graphs with bounded tree-width. Such result follows directly from [2] where problems closely related to communities were studied. Indeed, BALANCED 2-COMMUNITY corresponds to a t -Decomposition defined in [2] where the function $t = n/2$, the functions a, b are equal and for each vertex v in G let $a(v) = b(v) = \frac{n/2-1}{n-1}d(v)$, where n is the order of the graph. Since this problem was proved to be polynomial-time solvable for bounded tree-width in [2], we can conclude the same result for the BALANCED 2-COMMUNITY problem.

Now we focus on the problem of 2-communities in general graphs. In [7] it has been proved that to find a connected balanced partition without any additional constraints is an NP-complete problem in general graphs. We prove similar results for BALANCED WEAK 2-COMMUNITY and BALANCED 2-COMMUNITY and their connected variants. To show that BALANCED WEAK 2-COMMUNITY is NP-complete, we use the BALANCED CO-SATISFACTORY PARTITION problem which was proved NP-complete by Bazgan *et al.* [4]. The problem is defined as follow:

BALANCED CO-SATISFACTORY PARTITION

Input : A graph $G = (V, E)$ on an even number of vertices.

Question : Is there a balanced partition $\{C_1, C_2\}$ of V such that for every $v \in V$, $d_{in}(v) \leq d_{out}(v)$?

Theorem 4. BALANCED WEAK 2-COMMUNITY is NP-complete.

Proof. Clearly this problem is in NP. We reduce BALANCED CO-SATISFACTORY PARTITION to BALANCED WEAK 2-COMMUNITY. Let G be a graph on $2n$ vertices as an instance of BALANCED CO-SATISFACTORY PARTITION, and let G' , the complement of G , be an instance of BALANCED WEAK 2-COMMUNITY. If G admits a balanced co-satisfactory partition $\{C_1, C_2\}$ then $\{C_1, C_2\}$ is also a weak 2-community. Indeed, for every vertex $v \in V$, then $d_{in}(v) \leq d_{out}(v)$ in G . Thus, in G' we have $d_{in}^{G'}(v) = 2n - 1 - d_{in}(v) \geq 2n - 1 - d_{out}(v) = d_{out}^{G'}(v)$. Conversely, any balanced weak 2-community in G' is a balanced co-satisfactory partition in G . \square

Due to the construction of G' in Theorem 4 and the reduction from [4] we can conclude the following:

Theorem 5. CONNECTED BALANCED WEAK 2-COMMUNITY is NP-complete.

Estivill-Castro *et al.* [9] have shown that BALANCED 2-COMMUNITY is NP-complete by constructing a reduction from a variant of the CLIQUE problem. We propose a shorter alternative proof which is also valid for the CONNECTED BALANCED 2-COMMUNITY problem. The proof is based on the NP-complete problem BALANCED SATISFACTORY PARTITION which was introduced by Bazgan *et al.* [3] as follows:

BALANCED SATISFACTORY PARTITION

Input : A graph $G = (V, E)$ on an even number of vertices.

Question : Is there a balanced partition $\{C_1, C_2\}$ of V such that for every $v \in V$, $d_{in}(v) \geq \frac{d(v)}{2}$?

It can be proved that these two problems are in fact equivalent when the number of vertices is even.

Lemma 4. Let $G = (V, E)$ be a graph with n vertices. Consider a partition $\{C_1, C_2\}$ of V and $v \in C_1$. Then the following assertions are equivalent:

1. $\frac{d_{in}(v)}{|C_1|-1} \geq \frac{d(v)}{n-1}$
2. $\frac{d_{out}(v)}{|C_2|} \leq \frac{d(v)}{n-1}$
3. $\frac{d_{in}(v)}{|C_1|-1} \geq \frac{d_{out}(v)}{|C_2|}$

Remark. Notice that the third assertion in Lemma 4 is the condition (1) of a 2-community structure.

Lemma 5. Let $G = (V, E)$ be a graph with an even number n of vertices. Consider a balanced partition $\{C_1, C_2\}$ of V . Then for any vertex $v \in V$, $d_{in}(v) = \frac{|C_1|-1}{n-1}d(v)$ if and only if $d(v) = n-1$.

Proof. If $d(v) = n-1$, then clearly $d_{in}(v) = n/2 - 1$. Suppose now that $d_{in}(v) = \frac{n/2-1}{n-1}d(v)$. Notice that $(-2)(n/2 - 1) + 1(n-1) = 1$ from which it can be easily shown that $n/2 - 1$ and $n-1$ do not have common divisors. This implies that $d(v)$ is a multiple of $n-1$. Thus, $d(v) = n-1$. \square

Remark. Let $\{C_1, C_2\}$ be a balanced partition of G and $v \in C_1$ be a vertex of degree $n-1$. Since v has $\frac{n}{2} - 1$ neighbors in its own part and $\frac{n}{2}$ in other part, v does not satisfy the condition of BALANCED SATISFACTORY PARTITION. However, v satisfies the BALANCED 2-COMMUNITY condition since $\frac{d_{in}(v)}{|C_1|-1} = 1$.

Proposition 1. Let $G = (V, E)$ be a graph with n vertices without vertices of degree $n-1$. Then BALANCED SATISFACTORY PARTITION and BALANCED 2-COMMUNITY are equivalent on G .

Proof. Suppose that G is a yes-instance of BALANCED SATISFACTORY PARTITION, that is there exists a balanced partition $\{C_1, C_2\}$ of V such that any vertex $v \in V$ satisfies the condition $d_{in}(v) \geq \frac{1}{2}d(v)$, which implies that $d_{in}(v) \geq \frac{|C_1|-1}{2|C_1|-1}d(v) = \frac{|C_1|-1}{n-1}d(v)$. Thus, G is a yes-instance of BALANCED 2-COMMUNITY.

Suppose now that G is a yes-instance of BALANCED 2-COMMUNITY, that is there exists a balanced partition $\{C_1, C_2\}$ of V such that any vertex $v \in V$ satisfies the condition $d_{in}(v) \geq \frac{|C_1|-1}{|C_2|}d_{out}(v)$ that is equivalent to $d_{in}(v) \geq \frac{|C_1|-1}{n-1}d(v)$ using Lemma 4. According to Lemma 5, there is no vertex v such that $d_{in}(v) = \frac{|C_1|-1}{n-1}d(v)$.

We have to show that for every vertex $v \in V$, $d_{in}(v) \geq \frac{1}{2}d(v)$. Suppose by contradiction that there exists a vertex $v \in V$ that does not satisfy the inequality that is

$$\frac{|C_1|-1}{n-1}d(v) < d_{in}(v) < \frac{1}{2}d(v)$$

First, notice that $\frac{1}{2}d(v) - \frac{|C_1|-1}{n-1}d(v) = \frac{1}{2(n-1)}d(v) < 1$, which means that there is at most one whole number between $\frac{|C_1|-1}{n-1}d(v)$ and $\frac{1}{2}d(v)$.

Moreover, $d(v)$ cannot be even, since otherwise $d(v)/2$ would be a whole number and thus $d_{in}(v)$ could not be a whole number. Then $d(v)$ is odd and let $d(v) = 2p + 1$, $p \in \mathbb{N}$. We will arrive to a contradiction by showing that $p < d_{in}(v) < p + 1/2$. Notice that $d(v) < n - 1 \Rightarrow \frac{d(v)-1}{2} < \frac{|C_1|-1}{n-1}d(v)$ that implies $p < \frac{|C_1|-1}{n-1}d(v) < d_{in}(v)$. Then, we have necessarily $d_{in}(v) \geq \frac{1}{2}d(v)$ for every vertex $v \in V$, that is G is a yes-instance of BALANCED SATISFACTORY PARTITION. \square

BALANCED SATISFACTORY PARTITION has already been proved NP-complete in [3], even if both parts are required to be connected. Moreover, the reduction which is used to prove it does not construct a graph with vertices of degree $n - 1$. Thus we obtain a similar result as in [9] (the authors mentioned along the lines that the proof works also in connected case).

Theorem 6. CONNECTED BALANCED 2-COMMUNITY is NP-complete.

On the other hand, it is interesting to notice that there exist graphs in which all 2-community structures are balanced (see Fig. 4).

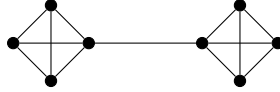


Fig. 4. A graph in which all 2-community structures are balanced

5 Conclusion and open problems

We studied the problems of existence and determination of a 2-community structure and its variants in graphs. We showed that every 3-regular graph has a 2-community structure and such a structure can be found in polynomial time. This

remains true even if connectivity of the partitions is required. The interesting open question is to determine if a graph of order at least 4 (except stars) has always a 2-community structure, even connected one. BALANCED 2-COMMUNITY is NP-complete in general graphs, but the complexity of determining a balanced 2-community structure in 3-regular graphs remains open. This last problem is equivalent to finding a cut whose edges forms a matching in 3-regular graphs.

For the weak version the situation is slightly different since any graph of maximum degree 3 has even a balanced weak 2-community structure that can be found in polynomial time. Furthermore, we proved that BALANCED WEAK 2-COMMUNITY is NP-complete on general graphs, even for connected communities. It remains open if any graph of order at least 4 has a weak 2-community structure, even connected one (except stars).

References

1. K. Andreev and H. Racke. Balanced graph partitioning. *Theory of Computing Systems*, 39(6):929–939, 2006.
2. C. Bazgan, Z. Tuza, and D. Vanderpooten. Degree-constrained decompositions of graphs: Bounded treewidth and planarity. *Theoretical Computer Science*, 355(3):389–395, 2006.
3. C. Bazgan, Z. Tuza, and D. Vanderpooten. The satisfactory partition problem. *Discrete Applied Mathematics*, 154(8):1236–1245, 2006.
4. C. Bazgan, Z. Tuza, and D. Vanderpooten. Approximation of satisfactory bisection problems. *Journal of Computer and System Sciences*, 74(5):875–883, 2008.
5. C. Bazgan, Z. Tuza, and D. Vanderpooten. Satisfactory graph partition, variants, and generalizations. *Europ. Journal of Operat. Research*, 206(2):271–280, 2010.
6. A. Buluç, H. Meyerhenke, I. Safro, P. Sanders, and C. Schulz. Recent advances in graph partitioning. *arXiv:1311.3144*.
7. J. Chlebikova. Approximating the maximally balanced connected partition problem in graphs. *Information Processing Letters*, 60(5):223–230, 1996.
8. D. Delling, A. Goldberg, T. Pajor, and R. Werneck. Customizable route planning. In *Proc. of 10th Intern. Symp. on Exper. Algor.*, LNCS 6630, pages 376–387, 2011.
9. V. Estivill-Castro and M. Parsa. On connected two communities. In *Proc. of the 36th Australasian Computer Science Conference (ACSC)*, pages 23–30, 2013.
10. A. E. Feldmann and L. Foschini. Balanced partitions of trees and applications. *Algorithmica*, 71(2):354–376, 2015.
11. S. Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, 2010.
12. M. E. J. Newman. Detecting community structure in networks. *The Europ. Physical Journal B-Condensed Matter and Complex Systems*, 38(2):321–330, 2004.
13. M. Olsen. A general view on computing communities. *Mathematical Social Sciences*, 66(3):331–336, 2013.
14. S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1:27–64, 2007.
15. D. B. Shmoys. Cut problems and their application to divide-and-conquer. In *Approximat. Algorithms for NP-Hard Problems*, PWS Publishing, 192–235, 1996.